Predicting and Controlling Resource Usage in a Heterogeneous Active Network

Virginie Galtier, Yannick Carlinet and Kevin L. Mills (NIST)
Stephen F. Bush and Amit B. Kulkarni (GE CRD)

National Research Council Review Meeting
February 9, 2001

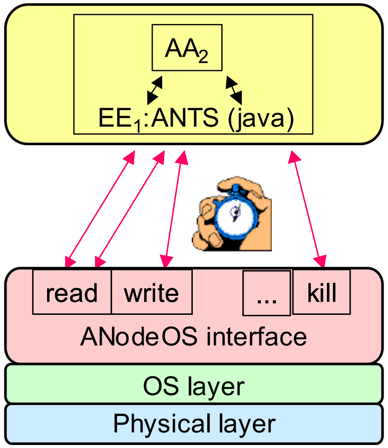
Overview of Presentation

- Modeling an Application's CPU-Time Use
- Adapting CPU-Time Models for Use by Mobile Code in Heterogeneous Networks
- Applying Adaptive CPU-Time Models **(Experiments in Progress)**
  - Control Execution of Mobile Code in Magician Execution Environment
  - Predict CPU Consumption among Network Nodes using GE's Active Virtual Network Management Prediction (AVNMP) System
- Future Research and Related Publications

Modeling CPU Use by Applications

NIST National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

(1) Monitor at System Calls in Node Operating System



(2) Generate Execution Trace

...

begin, user (4 cc), read (20 cc), user (18 cc),
write(56 cc), user (5 cc), end

begin, user (2 cc), read (21 cc), user (18 cc),
kill (6 cc), user (8 cc), end

begin, user (2 cc), read (15 cc), user (8 cc),
kill (5 cc), user (9 cc), end

begin, user (5 cc), read (20 cc), user (18 cc),
write(53 cc), user (5 cc), end

begin, user (2 cc), read (18 cc), user (17 cc),
kill (20 cc), user (8 cc), end

...

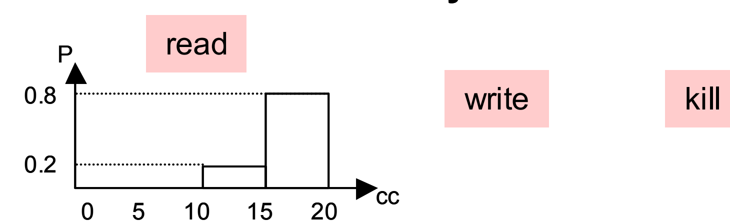
Trace is a series of system calls and transitions stamped with CPU time use

(3) Consume Trace & Generate Application Model

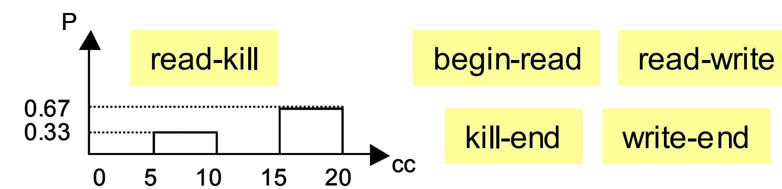
Scenario A:
sequence = "read-write",
probability = 2/5

Scenario B:
sequence = "read-kill",
probability = 3/5

Distributions of CPU time in system calls :



Distributions of CPU time between system calls :



NIST CENTENNIAL 1901-2001 DARPA

Each Node Constructs a Node Model using two calibration benchmarks:

- a system benchmark program ⌫ for each system call, average system time
- for VM, an app. benchmark program ⌫ average time spent in the VM between system calls

Scaling From Node X to Node Y*

App. model on node X:
```
read   30 cc
user   10 cc
write  20 cc
```

Model of node X:
```
read   40 cc
write  18 cc
user   13 cc
```

scale

Model of node Y:
```
read   20 cc
write  45 cc
user    9 cc
```

App. model on node Y:
```
read   30*20/40 = 15 cc
user   10*9/13  =  7 cc
write  20*45/18 = 50 cc
```

*To scale an App. Model in a network, select one Node Model as a reference known by all other nodes

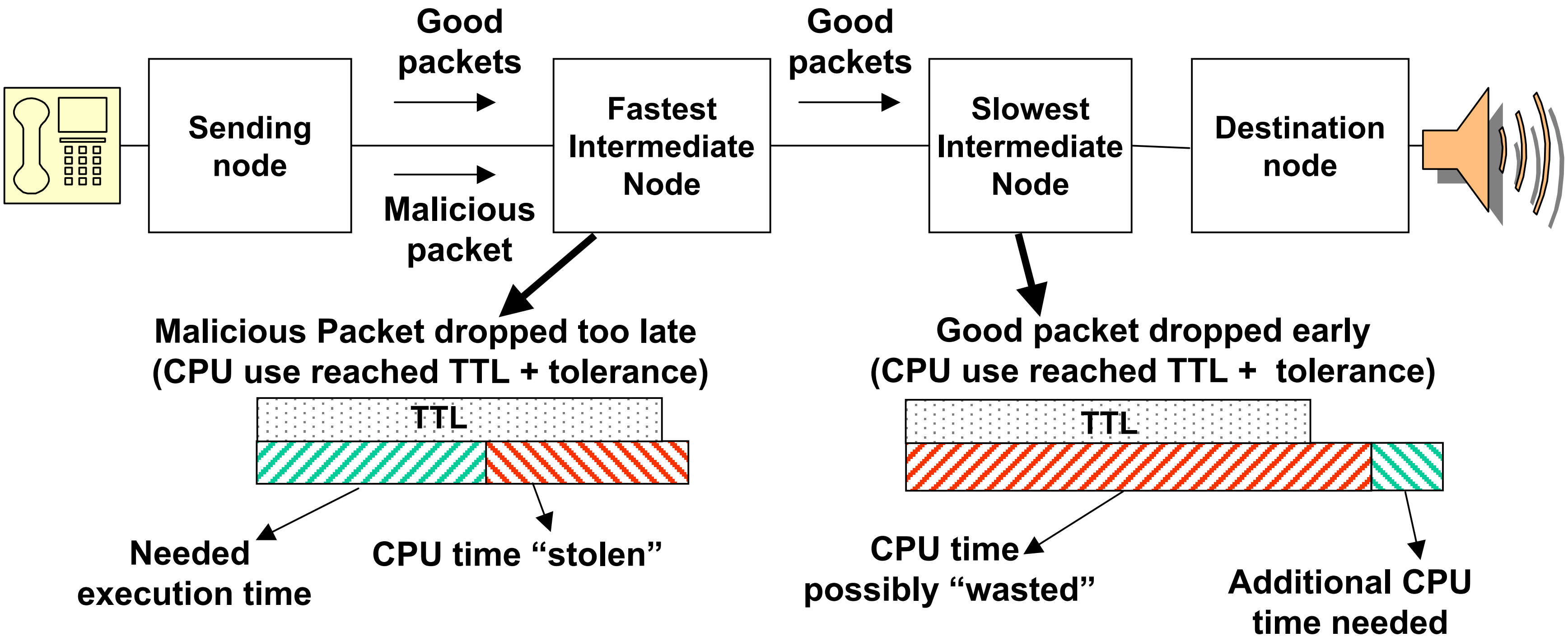
Some Sample Results: Scaling Magician Application Models between Selected Pairs of Nodes vs. Scaling with Processor Speeds Alone

			Scaling with Models		Scaling with Speeds	
AA	Node X	Node Y	Mean	Avg. High Per.	Mean	Avg. High Per.
Ping	Blue	Black	<1	21	15	38
	Blue	Green	2	18	13	15
	Black	Blue	<1	16	13	25
	Red	Green	6	10	92	82
	Red	Black	4	14	154	135
	Yellow	Black	6	16	190	163
	Yellow	Green	8	15	119	103
	Black	Green	4	23	24	22
Route	Blue	Black	2	9	15	250
	Black	Blue	<1	23	13	32
	Red	Green	4	15	88	64
	Red	Black	6	19	155	137
	Yellow	Black	5	16	190	164
	Yellow	Green	6	14	114	83
	Black	Green	3	28	26	28
	Blue	Green	<1	28	15	204

The Average Absolute Deviation (in Percent) of Simulated Predictions from Measured Reality for Each of Two Active Applications (Average High Percentile Considers Combined Comparison of 80th, 85th 90th, 95th, and 99th Percentiles)

Application: Control Execution of Mobile Code

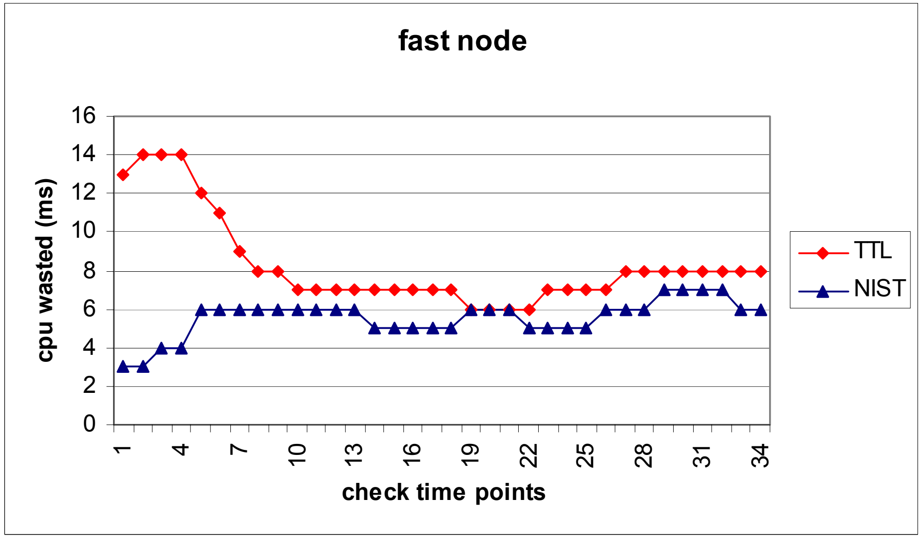
Experiment in Progress: Control CPU Usage by Mobile Programs

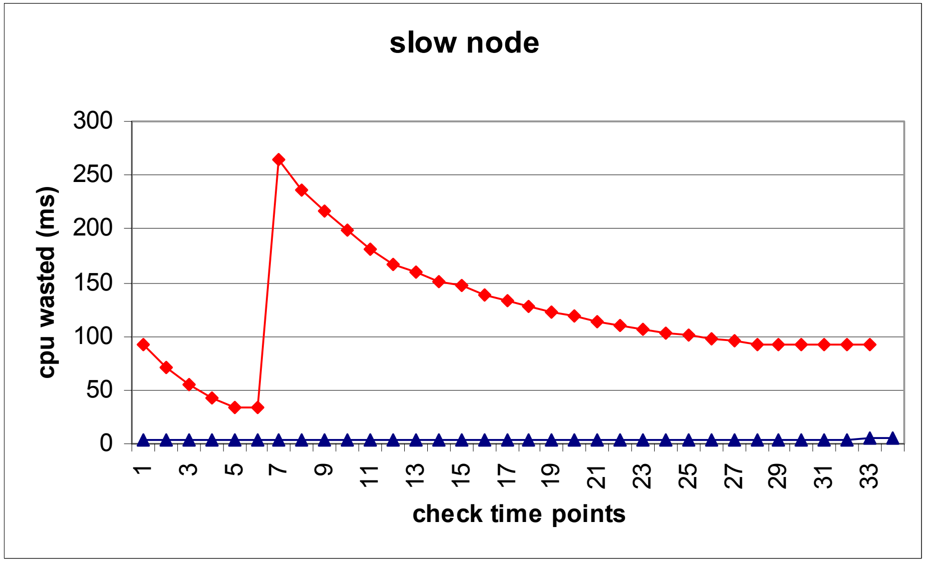


When mobile code CPU usage controlled with fixed allocation or TTL, malicious or "buggy" mobile programs can "steal" substantial CPU cycles, especially on fast nodes

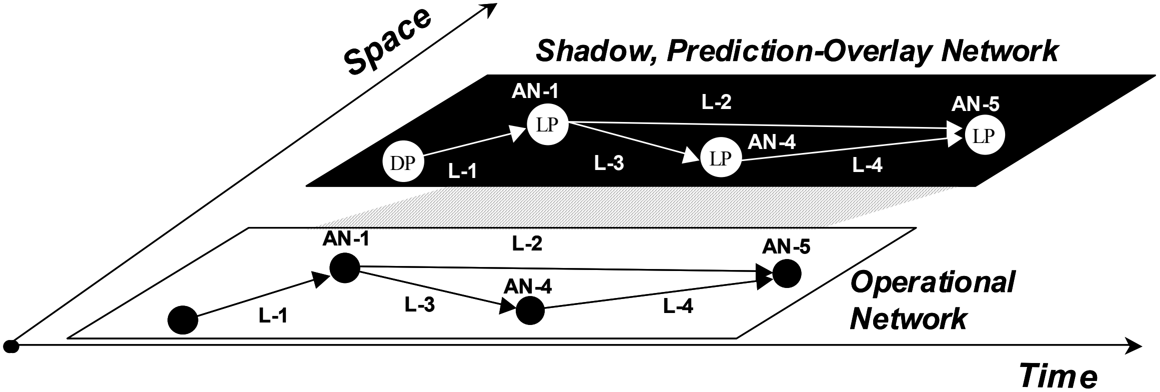
When mobile code CPU usage controlled with fixed allocation or TTL, correctly coded mobile programs can be terminated too soon on slow nodes, wasting substantial CPU cycles

CPU Control: Expected Results

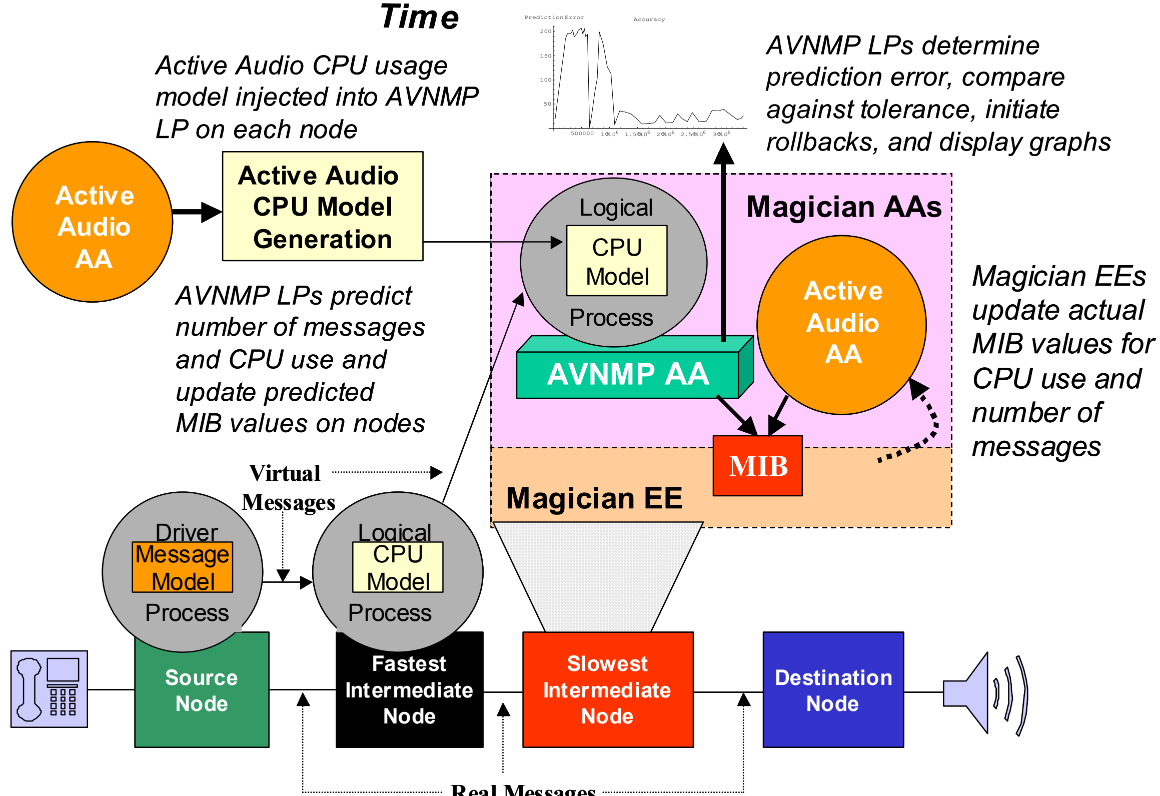




Application: Predict CPU Usage among Network Nodes: Experiment in Progress



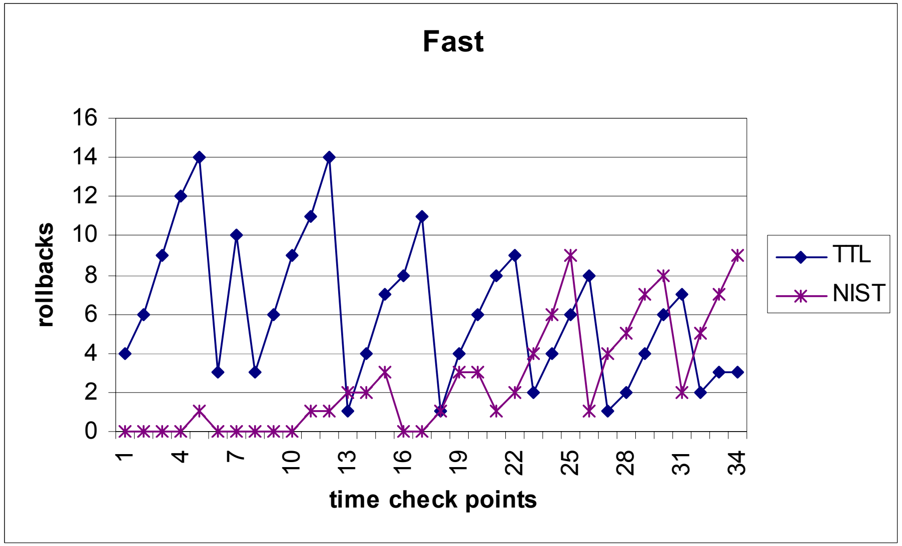
GE Active Virtual Network Management Prediction (AVNMP) System

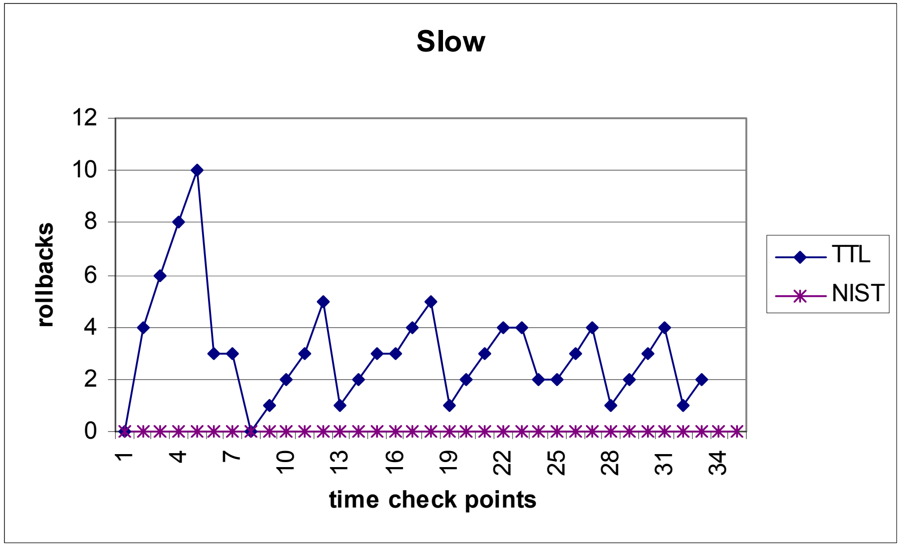


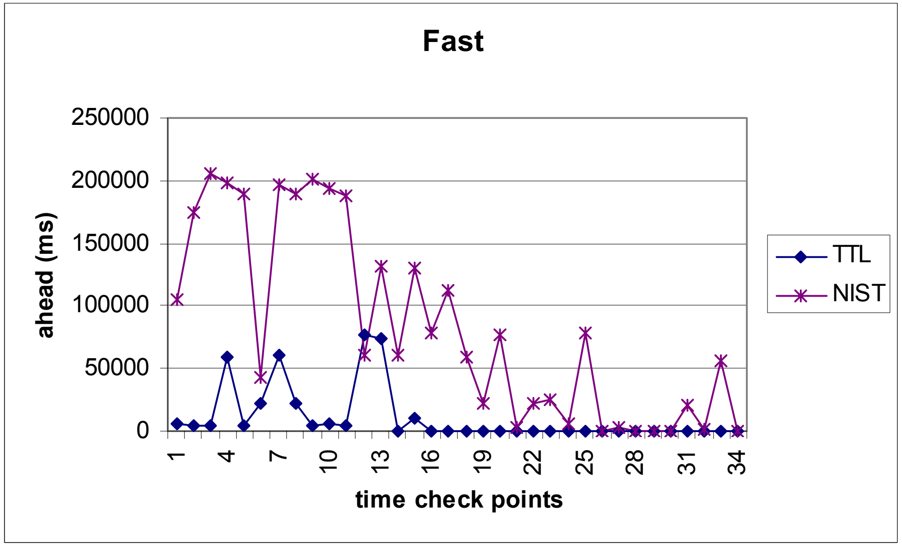
Can NIST Models enable AVNMP to predict CPU use among heterogeneous network nodes, while providing better look ahead and improved prediction efficiency than simple TTL approaches?

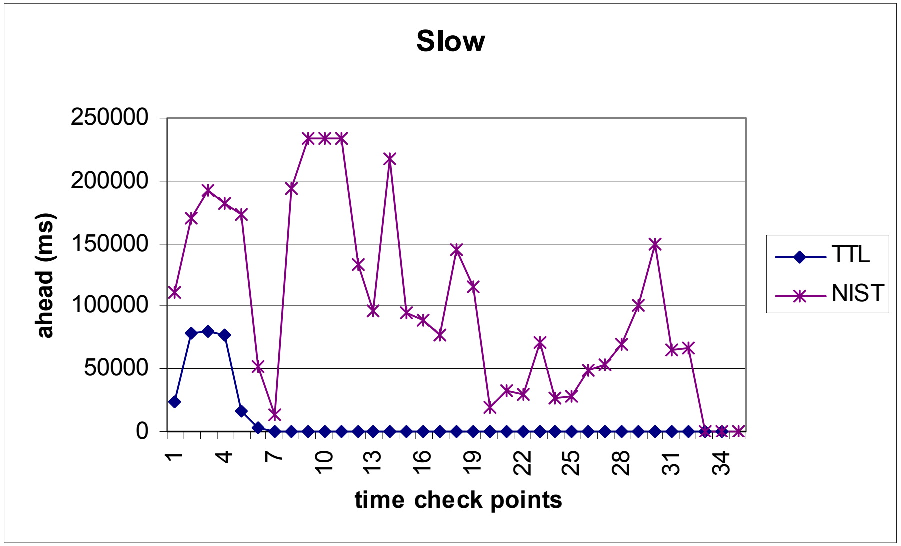
CPU Prediction: Expected Results

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce









Future Research

- Improve Our Models
  - Model Node-Dependent Conditions
  - Attempt to Characterize Errors Bounds
  - Improve the Space-Time Efficiency of Our Models
  - Continue Search for Low-Complexity Analytically Tractable Models
  - Investigate Models that Continue to Learn

- Investigate Competitive-Prediction Approaches
  - Run Competing Predictors for Each Application
  - Score Predictions from Each Model and Reinforce Good Predictors
  - Use Prediction from Best Scoring Model

- Apply Our Models
  - CPU Resource Allocation Control in Node Operating System
  - Network Path Selection Mechanisms that Consider CPU Requirements
  - CPU Resource Management Algorithms Distributed Across Nodes

Related Publications

V. Galtier, C. Hunt, S. Leigh, K. Mills, D. Montgomery, M. Ranganathan, A. Rukhin, and D. Tang, "How Much CPU Time?", *Draft NIST Technical Report*, TR-ANTD-ANETS-111999, November 1999. <http://w3.antd.nist.gov/~mills/unpublished/NISTanetsTR.pdf>

Y. Carlinet, V. Galtier, K. Mills, S. Leigh, A. Rukhin, "Calibrating an Active Network Node," *Proceedings of the 2nd Workshop on Active Middleware Services*, ACM, August 2000. <http://w3.antd.nist.gov/~mills/papers/Final-woasm.pdf>

V. Galtier, K. Mills, Y. Carlinet, S. Leigh, A. Rukhin, "Expressing Meaningful Processing Requirements among Heterogeneous Nodes in an Active Network," *Proceedings of the 2nd International Workshop on Software Performance*, ACM, September 2000. <http://w3.antd.nist.gov/~mills/papers/WSOPfu-04.pdf>

V. Galtier, K. Mills, Y. Carlinet, S. Bush, and A. Kulkarni, "Predicting Resource Demand in Heterogeneous Active Networks", submitted to *MILCOM 2001*. <http://w3.antd.nist.gov/~mills/unpublished/ALTmilcom2001v4.pdf>